

Harmonizer

Audrey Cooke, Michal Courson, Daniel Kohler, Allanah Matthews

A dark blue, diagonal shape that starts from the bottom left corner and extends towards the top right, covering the lower half of the slide.

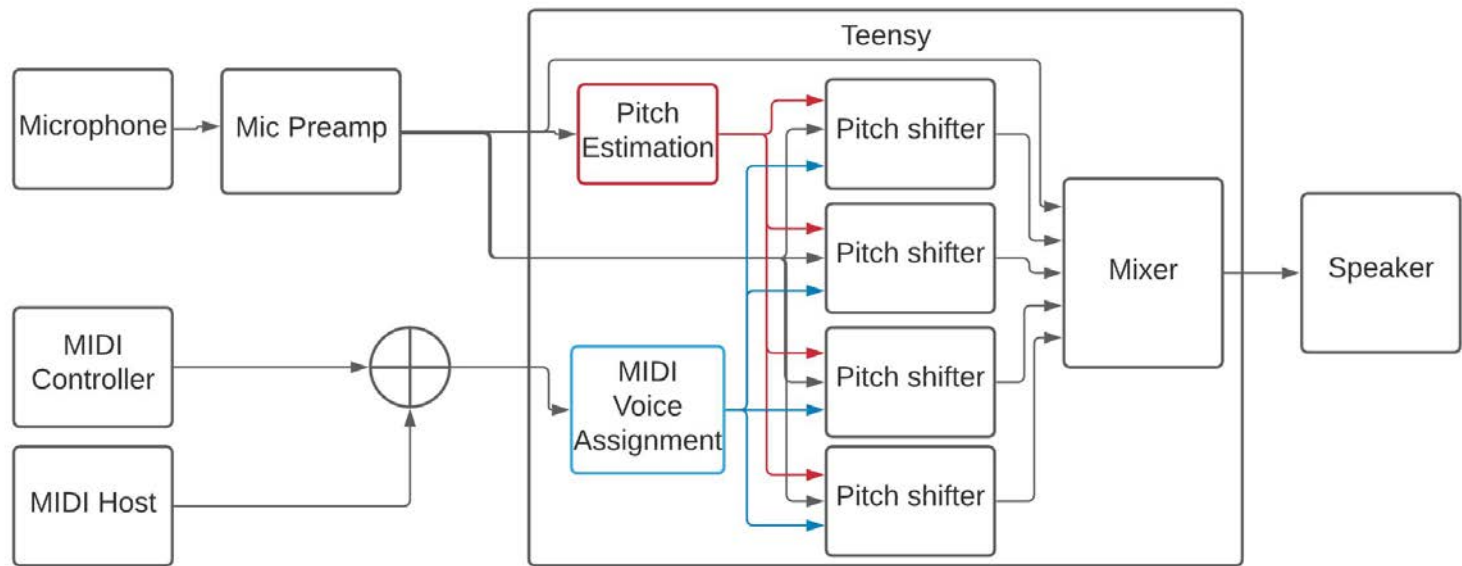
Background

- The Harmonizer is a device to pitch shift monophonic input signals to user defined frequencies and mix these pitch shifted results to create a harmonious output.
- Using the Harmonizer, a single musician can create harmony without the dependence for accompanying musicians!

Past Work

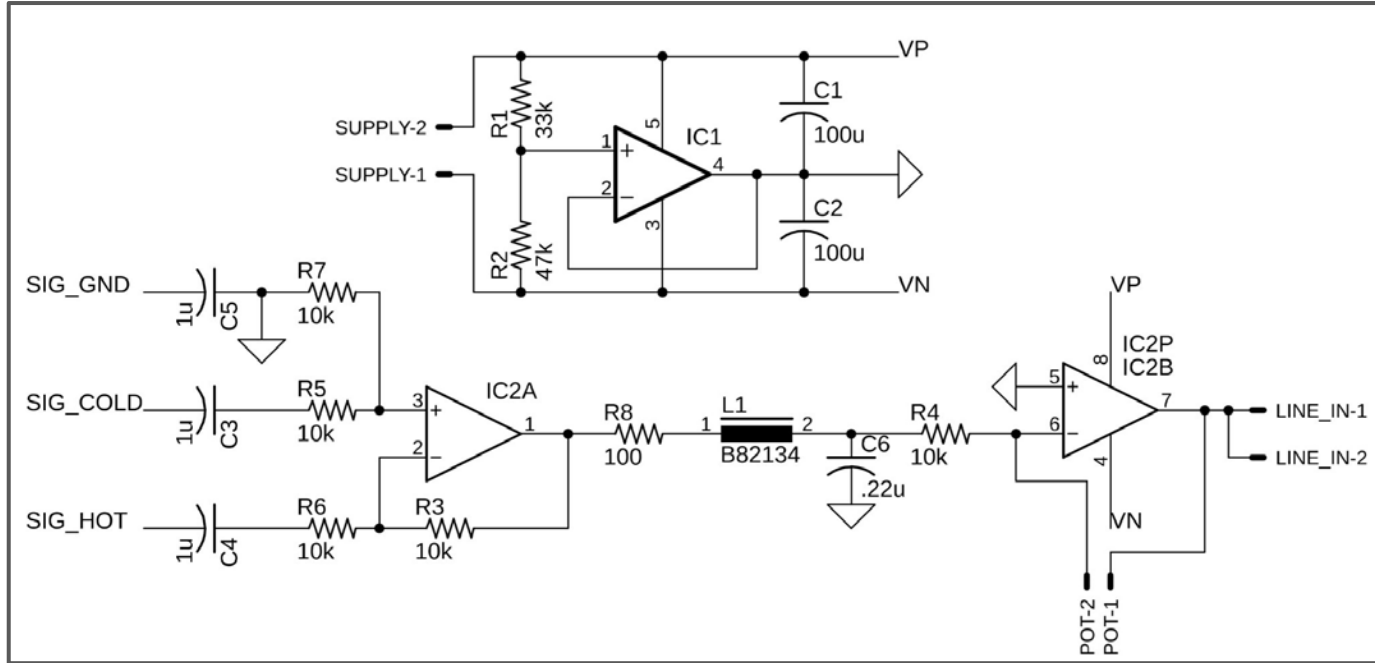
- Jacob Collier's Harmonizer -- PC with Ableton, dismantled TC Helicon, custom code on DSP hardware
- Bon Iver's Messina -- not a ton publicly known, Ableton does some preprocessing, sent to Eventide H8000 for pitch shifting with a proprietary algorithm
- Getting algorithms that sound "perfect" is difficult

High-Level Description



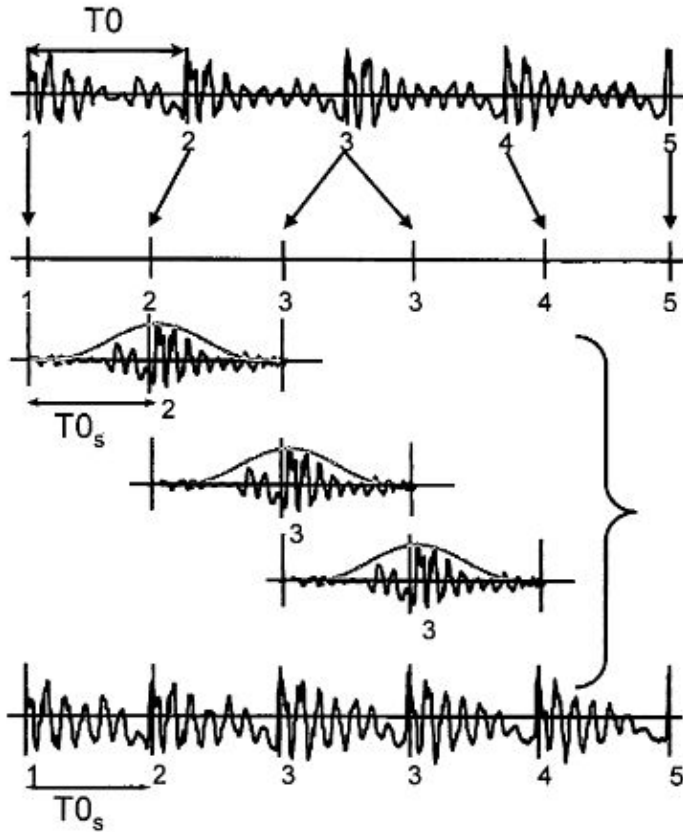
Hardware

Microphone



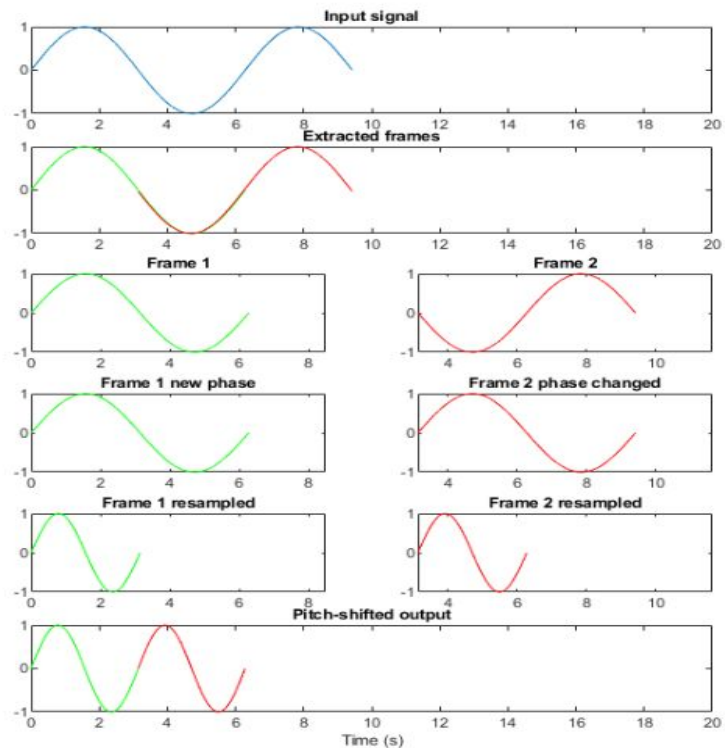
Teensy 4.1 and
Audio Shield

TD-PSOLA



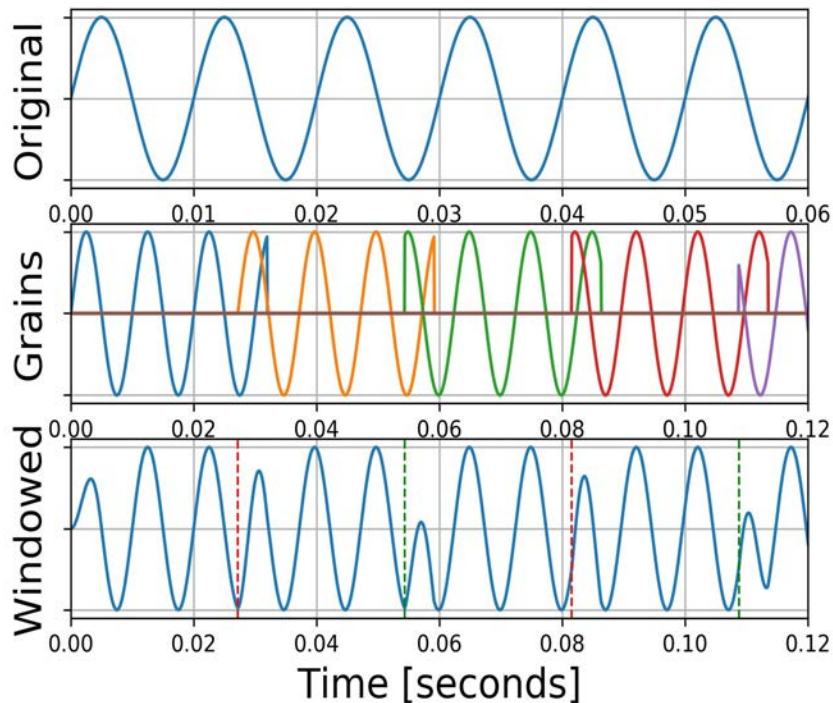
- Generate original signal pitch markings
- Calculate spacing of new pitch markings
- Window two waveforms of original signal
- Overlap and add windowed waveforms aligned with new pitch markings

Phase-Vocoding



- Perform RFFT on adjacent windows of input
 - Separated by analysis hopsize
- Perform phase shift on each RFFT window
- Perform IRFFT
- Perform overlap and add
 - Separated by synthesis hopsize
- Resample in order to perform pitch shift

Granular Synthesis



- Segment sections of original signal
- Time stretch by copying and overlapping segments
- Use envelopes to smooth discontinuities
- Resample to alter pitch

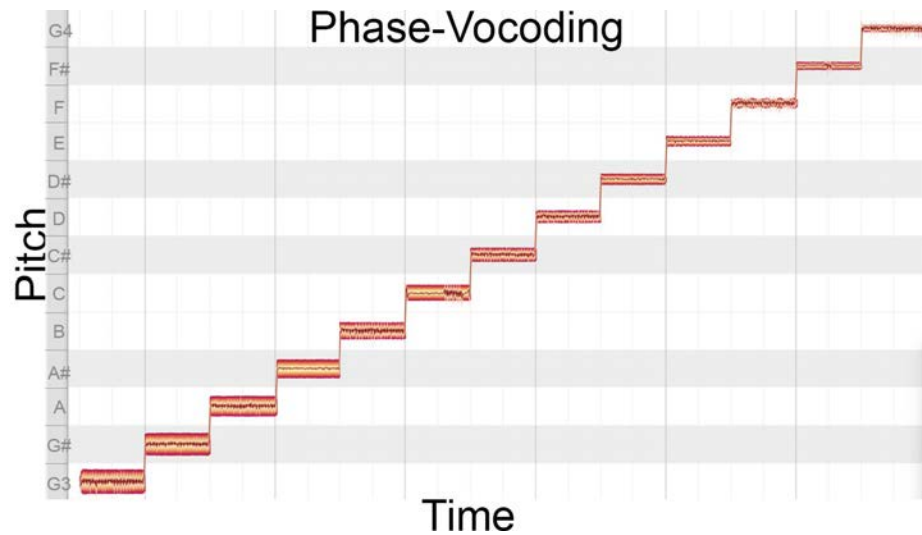
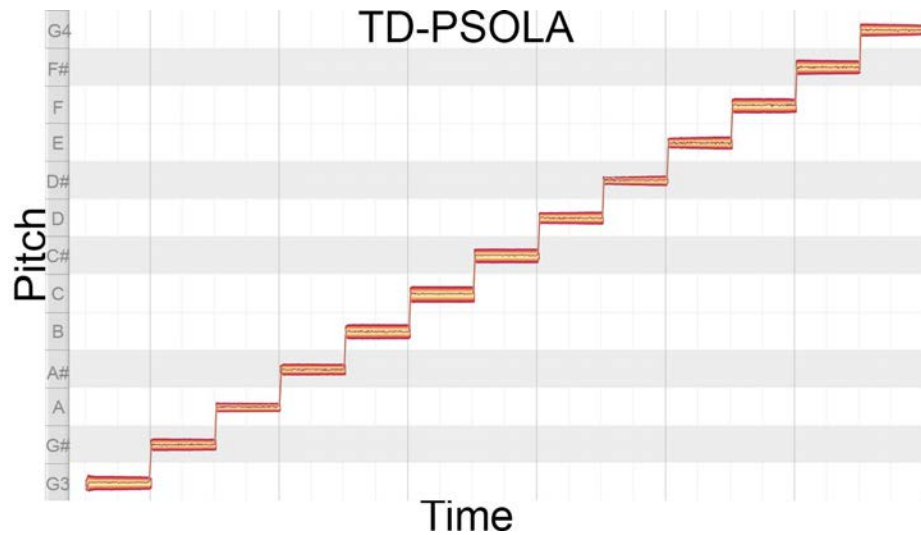
Technical Issues

- Phase Vocoding
 - Difficulty translating to C++ offline version due to differences in FFT implementations with Python
 - Could not meet timing requirements on first iteration of real-time implementation
 - Had to switch FFT algorithms to meet timing requirement
- TD-PSOLA
 - Pitch shifting to lower notes does not sound very good
- Granular Synthesis
 - Output sound is not smooth
 - Discontinuities in grains
 - Frequency alias based on grain size

Testing

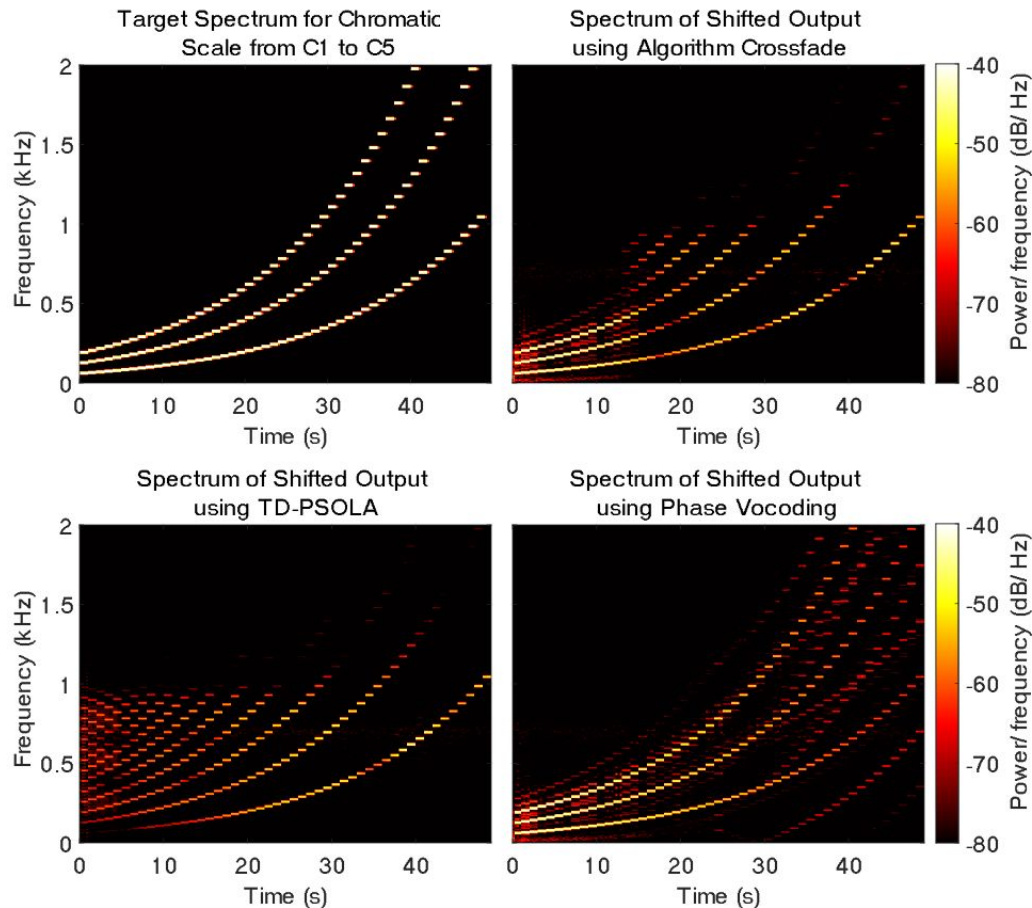
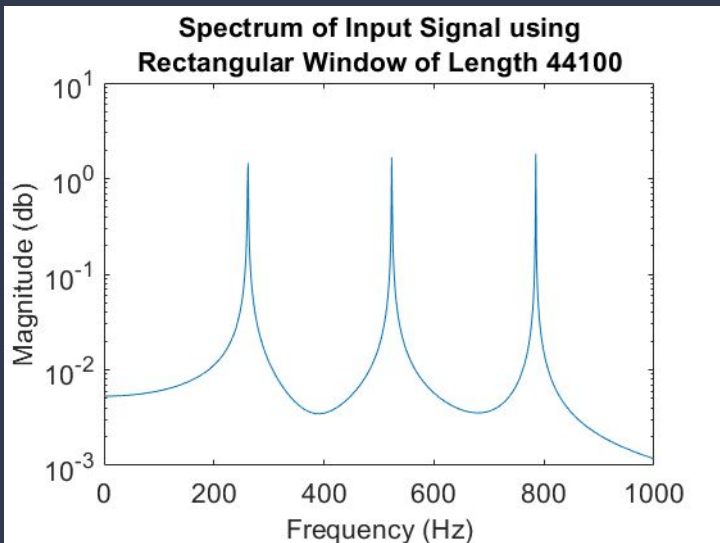
- Monophonic - testing how well target pitch is hit
 - Shift to every semitone +/- an octave
 - Tuner should register each note as in tune
- Polyphonic - testing how "good" the algorithms sound
 - Subjective timbre preferences
- Timing - testing how fast the algorithms run
 - Arduino micros() function
 - Must run fast enough to process before next block arrives

TD-PSOLA Tuning Results

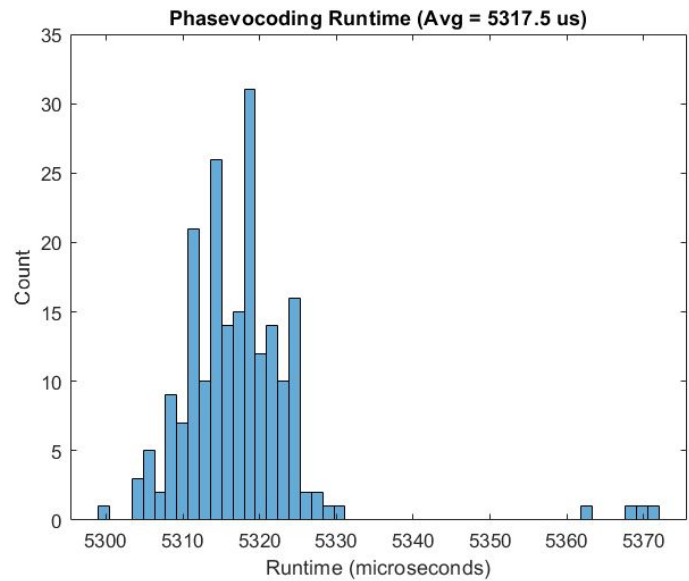
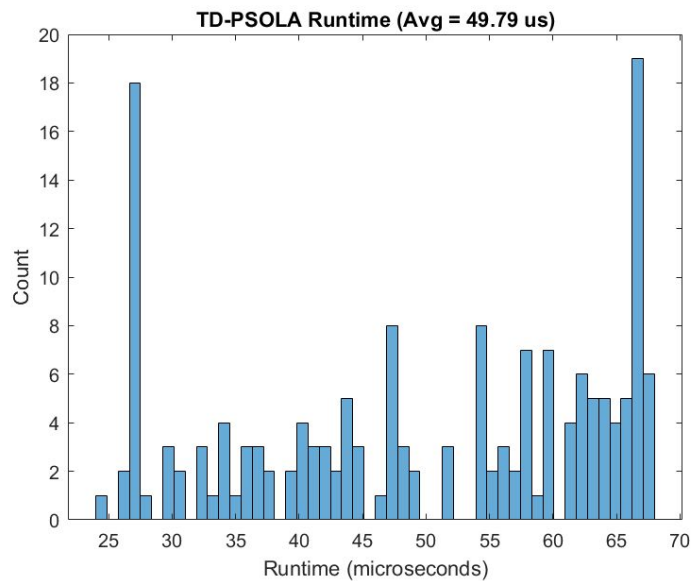


Algorithm Switching

- TD-PSOLA has frequency artifacts for low shifts
- Phase-Vocoding has greater artifacting for high shifts
- Using a mix of the two produces decent results



Timing Results



Results – Demo 1



Hide_and_Seek.mp4 in videos folder

Results – Demo 2



sax_improv.mp4 in videos folder

Future Work

- Minimize latency
 - Analysis algorithm optimization
- More concurrent voices
 - Modify TD-PSOLA for low shifts
 - Speed up Phase Vocoder
- Controllable effects
 - Reverb
 - Filters
 - Delays
 - Portamento
 - Windowing
- Noise reduction
 - Minimize ground loop interference
 - Isolate audio hardware from power hardware